



## Aplicando polimorfismo

### Transcripción

[00:00] Vamos a presentar nuestro segundo ejemplo y es también un caso muy, muy, muy típico. Recuerdan que nosotros tenemos la bonificación de nuestros trabajadores. Es casi seguro que nosotros vamos a necesitar un control de estas bonificaciones, vamos a necesitar saber cuál es la bonificación total que estamos dando como empresa, como banco y para eso vamos a crear una clase control de nuestras bonificaciones.

[00:30] Esto sería ControlBonificacion. Perfecto. En esta clase, ¿qué es lo que vamos a hacer? En esta clase nosotros vamos a sumar todas las bonificaciones que le vayamos dando como parámetro. Para esto vamos a definir nuestra primera variable que va a ser un tipo privado, un entero. Mejor un double, que se va a llamar suma total, es la suma de bonificaciones.

[01:05] Y aquí un método, ese sí va a ser public, que va a retornar un double, llamada registrar salario. ¿Por qué? Porque basado en el salario es que se obtiene la bonificación, ¿y qué va a recibir como parámetro? Como parámetro en este caso vamos a recibir a un funcionario, porque estamos acumulando el salario del funcionario.

[01:38] El método no compila porque como es un tipo double press necesitamos retornar un tipo de dato y esto sería como decirle que la suma, this.suma, es igual a el salario del funcionario, perdón, es la bonificación del funcionario más, como estamos sumando la de todos, this.suma. Perfecto.

[02:10] De modo que yo voy a ir sumando poco a poco aquí, y cada vez que yo registre a un nuevo funcionario entonces yo voy a obtener la bonificación y la voy a sumar a la suma actual que yo ya tengo acumulada. Perfecto. ¿Qué vamos a retornar? Vamos a retornar `this.suma`. Sería un `return this.suma`. Listo. No hay nada mágico hasta ahí, no hay nada nuevo, es un código ya conocido, un método ya conocido.

[02:50] Entonces ahora vamos a crear un nuevo test. Creamos una nueva clase y la vamos a llamar `TestControlBonificacion`. Listo. Le damos finish. Nuestro ya superconocidísimo método `main`, e igual que en el ejemplo anterior, vamos a crear un funcionario, funcionario vamos a llamar ahora Diego igual `new funcionario`, y a este funcionario le vamos a asignar un salario, `setSalario`, que va a ser en este caso 2000, igualito.

[03:36] Y ahora vamos a crear un gerente, vamos a llamarlo Jimena otra vez, para que sean nuestros dos personajes que ya estamos poniendo aquí en este contexto. Igualito `new gerente`. Hasta ahora nada nuevo. Y nuevamente `salario Jimena.setSalario` que ahora va a ser igual 10000. Listo. Hasta aquí nada nuevo. Y ahora mi control de bonificación. Guardo todo primero.

[04:14] Y ahora abro `controlBonificación`, "Ctrl + espacio", me dice ahí, `controlBonificacion`, ¿y esto va a ser igual a qué? `new controlBonificacion`. Listo. Incluso como aquí ya está quedando un poco largo yo le voy a dar aquí un salto de línea para que entre a la pantalla. Correcto. Entonces, aquí yo le voy a decir que registre `controlBonificacion.registrarSalario` y le vamos a mandar ¿a quién?

[04:56] A Diego, porque Diego es un funcionario. Entonces, aquí ya él ya está registrando el salario de Diego y vamos a decirle para tener un resultado para comparar, que antes de retornar la suma, que me imprima la suma, hasta dónde va. Perfecto. `Print this.suma`. ¿Para qué? Para yo ver en la pantalla cuál es el resultado actual de este cálculo.

[05:30] Incluso le voy a poner adelante cálculo actual, dos puntos, y la suma que yo voy acumulando hasta el momento. Perfecto. Guardamos, venimos aquí, ejecutamos nuestro método main, damos play y en el resultado cálculo actual 100. Y ahora, ese cálculo, ¿está bien hecho? Sí, está bien hecho porque recuerden que quedamos que iba a ser el 5% si no me equivoco, déjenme confirmar aquí, la bonificación del funcionario.

[06:22] Sí. 5% de su salario iba a ser su bonificación. Entonces, aquí en el TestControlBonificacion, el cálculo está bien hecho. Perfecto. Pero claro, es el único que tenemos. Y ahora, si necesitamos tener el control de bonificación del gerente, si queremos registrar al gerente, ¿cómo podemos hacer? Bueno, vamos a nuestra clase controlBonificacion.

[06:58] Y si aquí estamos aceptando a un funcionario, entonces aquí vamos a darle un "Ctrl + C", regresamos y aquí le decimos que en este caso va a ser un gerente. ¿Qué les parece? Y vemos que compila correctamente, acá le cambiamos para gerente, porque él ya está aceptando gerentes, "Ctrl + C", "Ctrl + V" para que compile y listo. Ya estamos, al mismo objeto suma le estamos sumando la bonificación del gerente.

[07:33] Entonces ahora vamos nuevamente al TestControl de bonificación. Yo voy a cerrar estas clases solamente para tener más espacio aquí. Listo. Y mismo ejercicio. ControlBonificacion.registrarSalario. ¿De quién? De gerente. Ahora tengo habilitado aquí el gerente, perfecto Aquí le puedo decir que registré el salario de Jimena. Ahora, si yo ejecuto esta clase vamos a ver cuánto va a calcular.

[08:08] Primer cálculo en 100, segundo cálculo 10600. ¿Esto por qué es? Porque está sumando el salario del gerente, porque recuerden que la bonificación del gerente es el salario completo, que ya lo tiene acá, más los 100 que tenía acá más un 5%. Perfecto. ¿Por qué? Vamos a la clase gerente para ver cómo es su regla de bonificación. Y en efecto es eso, es su salario más la bonificación actual.

[08:42] El 5% de 10000 serían 500. 10000, que es el salario, más 500, serían 10500, más 100 que ya le hemos bonificado al funcionario, 10600. Entonces los cálculos están bien hechos, están registrando bien. Y nuevamente comenzamos aquí a registrar un patrón no muy amigable. Vamos a hacerlo un poco más obvio.

[09:08] Supongamos que ahora yo no solamente tengo, ya habíamos quedado que en un banco tú tienes gerentes, tienes funcionarios, tienes contadores, tienes diseñadores, etcétera. Vamos a crear una nueva clase llamada Contador. Y este contador también es un funcionario, entonces también extends Funcionario. Hasta aquí nada nuevo.

[09:42] A modo de pregunta, de consulta, a modo de provocación. ¿Ustedes qué problema creen que estamos comenzando a tener aquí en este punto? Yo estoy creando un nuevo objeto, un nuevo tipo de funcionario, y vemos cómo está quedando mi clase de aquí. Hagan ese ejercicio, intenten ver aquí qué problema estamos ya comenzando a tener aquí con este código.